



UNITED STATES PATENT AND TRADEMARK OFFICE

98
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/707,471	12/16/2003	Rajesh Chawla	SYB/0093.01	1470
31779	7590	05/03/2006	EXAMINER	
JOHN A. SMART 708 BLOSSOM HILL RD., #201 LOS GATOS, CA 95032-3503			LOVEL, KIMBERLY M	
			ART UNIT	PAPER NUMBER
			2167	

DATE MAILED: 05/03/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/707,471	CHAWLA ET AL.	
	Examiner Kimberly Lovel	Art Unit 2167	

— The MAILING DATE of this communication appears on the cover sheet with the correspondence address —

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 16 December 2003.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-55 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-55 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 16 December 2003 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
 Paper No(s)/Mail Date 1/05/2005.

4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____.
 5) Notice of Informal Patent Application (PTO-152)
 6) Other: _____.

DETAILED ACTION

1. Claims 1-55 are rejected.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on 5 January 2005 was filed after the mailing date of the application on 16 December 2003. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Drawings

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: Fig. 5A, item 500. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-39 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

MPEP 2106 IV.B.2.(b)

A claim that requires one or more acts to be performed defines a process. However, not all processes are statutory under 35 U.S.C. 101. Schrader, 22 F.3d at 296, 30 USPQ2d at 1460. To be statutory, a claimed computer-related process must either: (A) result in a physical transformation outside the computer for which a practical application is either disclosed in the specification or would have been known to a skilled artisan, or (B) be limited to a practical application.

Claim 1 recites a method for performing database operations on data obtained from a web service, the method comprising: creating at least one proxy table in a database, each proxy table mapping to a method of the web service; in response to a database operation on a particular proxy table, converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping; invoking the particular method of the web service; converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping; and performing the database operation on the data at the database.

In the above limitation, there is no physical transformation being claimed, a practical application would be established by a useful, concrete and tangible result.

For the result to be tangible, it must be more than a thought or a computation and must have a real world value rather than being an abstract idea. The invention as recited in the claim consists performing an operation on the database. It is unclear to as what kind of tangible output is obtained by these limitations. Claims 2-21 are dependent on the method of claim 1, and therefore are rejected on the same grounds as claim 1.

Claim 22 recites a system for performing operations at a database on data obtained from a remote service, the system comprising: a mapping module for creating database tables representing at least some methods of a remote service accessed through a defined interface; an invocation module for converting a database operation on a database table representing a method of the remote service into a call for invoking the method; a communication module for transmitting the call for invoking the method to the remote service, and returning result values from invoking the method to the database; and a conversion module for converting result values received from the method into database format.

Even though claim 22 recites a system, the claim is directed towards software per se. Software per se fails to produce a tangible result. In order for the subject matter to be considered tangible, it must produce a useful, concrete

and tangible result. Claims 23-39 are dependent on the system of claim 22, and therefore are rejected on the same grounds as claim 22.

To allow for compact prosecution, the examiner will apply prior art to these claims as best understood, with the assumption that applicant will amend to overcome the stated 101 rejections.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. Claims 1-55 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6,799,182 to Bata (hereafter Bata) in view of US PGPub 2003/0126136 to Omoigui (hereafter Omoigui).

Referring to claim 1, Bata teaches a method for performing database operations on data obtained from a web service. In particular, Bata teaches a method for performing database operations on data obtained from a web service (see abstract and column 4, lines 42-46), the method comprising:

creating at least one proxy table in a database, each proxy table mapping to a method of the web service (see column 5, lines 31-34 and column 9, lines 6-34 – the purchase order is stored in a database, which was generated from the input XML document);

in response to a database operation on a particular proxy table, converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping (see column 7, lines 35-43);

invoking the particular method of the web service (see column 10, lines 10-16 – the data is inserted);

converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping (see column 7, lines 44-46); and

performing the database operation on the data at the database (see column 9, lines 58-65 – the operation is a join operation to create the database).

However, while Bata discloses a method for performing database operations on data obtained from a web service, Bata fails to explicitly teach the

further limitation of performing the database operation on the data at the database. Omoigui teaches a similar method to that of Bata, including performing database operations on data (see [0498]-[0506]).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize Omoigui method of performing operations on the database created by Bata. One would have been motivated to do so in order to provide results from queries to a user.

Referring to claim 2, the combination of Bata and Omoigui (hereafter Bata/Omoigui) discloses the method of claim 1, wherein the web service comprises a service remotely available via a network (Bata: see column 4, lines 33-53 and Fig 3 – the service is available via the internet).

Referring to claim 3, Bata/Omoigui discloses the method of claim 1, wherein the web service has a Web Services Description Language (WSDL) interface (Omoigui: see [0467], lines 8-10 and [0532], lines 1-2).

Referring to claim 4, Bata/Omoigui discloses the method of claim 3, wherein said creating step includes creating said at least one proxy table based upon the WSDL interface (Omoigui: see [0467], lines 8-10 and [0532], lines 1-2).

Referring to claim 5, Bata/Omoigui discloses the method of claim 3, wherein said creating step includes substeps of:

obtaining the WSDL interface from the web service (Omoigui: see [0467], lines 8-10); and

creating said at least one proxy table based upon the WSDL interface (Omoigui: see [0532], lines 1-2).

Referring to claim 6, Bata/Omoigui discloses the method of claim 1, wherein said creating step includes creating meta data identifying a particular method of the web service to be invoked when a database operation is received on a particular proxy table (Omoigui: see [0736]).

Referring to claim 7, Bata/Omoigui discloses the method of claim 1, wherein said creating step includes mapping arguments of the method to fields of the proxy table (Bata: see column 9, lines 48-57 and Fig 9).

Referring to claim 8, Bata/Omoigui discloses the method of claim 1, wherein said creating step includes mapping arguments of the method to equivalent database data types (Bata: see column 9, lines 6-14).

Referring to claim 9, Bata/Omoigui discloses the method of claim 1, wherein said creating step includes creating an object encapsulating the mapping of a web method to the database (Bata: see column 6, lines 31-34).

Referring to claim 10, Bata/Omoigui discloses the method of claim 1, wherein said creating step includes storing the mapping between said at least one proxy table and methods of the web service (Bata: see column 6, lines 29-57 – the mapping is saved in a file folder).

Referring to claim 11, Bata/Omoigui discloses the method of claim 10, wherein said step of converting results includes consulting the mapping for converting the results into data for application at the database (Bata: see column 6, lines 29-57 – the mapping is saved in a file folder).

Referring to claim 12, Bata/Omoigui discloses the method of claim 1, wherein the database operation includes a selected one of a SELECT operation,

an INSERT operation, a JOIN operation, and a UNION operation (Omoigui: see [0287], lines 12-15 – SELECT operation).

Referring to claim 13, Bata/Omoigui discloses the method of claim 1, wherein said step of converting the database operation includes binding data from the database operation to a Simple Object Access Protocol (SOAP) call for invoking the particular method of the web service (Omoigui: see [0312] and [0362]).

Referring to claim 14, Bata/Omoigui discloses the method of claim 1, wherein said step of converting the database operation includes converting data from the database operation into Extensible Markup Language (XML) format (Bata: see column 3, lines 9-15).

Referring to claim 15, Bata/Omoigui discloses the method of claim 1, wherein said step of converting the database operation includes creating a Simple Object Access Protocol (SOAP) request for invoking the particular method of the web service (Omoigui: see [0312] and [0362]).

Referring to claim 16, Bata/Omoigui discloses the method of claim 15, wherein said step of invoking the particular method includes transmitting the SOAP request to a remote web service (Omoigui: see [0312] and [0362]).

Referring to claim 17, Bata/Omoigui discloses the method of claim 1, wherein said step of invoking the particular method includes receiving results from the web service (Omoigui: see [0080] and Fig 5).

Referring to claim 18, Bata/Omoigui discloses the method of claim 1, wherein said step of converting results includes converting results received in Simple Object Access Protocol (SOAP) format (Omoigui: see [0312] and [0362]).

Referring to claim 19, Bata/Omoigui discloses the method of claim 1, wherein said step of converting results includes converting results received in Extensible Markup Language (XML) format (Bata: see column 3, lines 9-15 and Fig 9).

Referring to claim 20, Bata/Omoigui discloses a computer-readable medium having processor-executable instructions for performing the method of claim 1 (Omoigui: see [0275]).

Referring to claim 21, Bata/Omoigui discloses a downloadable set of processor-executable instructions for performing the method of claim 1 (Omoigui: see [0275]).

Referring to claim 22, Bata teaches a system for performing operations at a database on data obtained from a remote service. In particular, system for performing operations at a database on data obtained from a remote service (see abstract), the system comprising:

a mapping module for creating database tables representing at least some methods of a remote service accessed through a defined interface (see column 9, lines 6-34);

an invocation module for converting a database operation on a database table representing a method of the remote service into a call for invoking the method (see column 10, lines 10-16);

a communication module for transmitting the call for invoking the method to the remote service, and returning result values from invoking the method to the database (see column 7, lines 35-43); and

a conversion module for converting result values received from the method into database format (see column 7, lines 44-46).

However, while Bata discloses a method for performing database operations on data obtained from a web service, Bata fails to explicitly teach the further limitation of returning result values. Omoigui teaches a similar method to that of Bata, including returning result values (see [0080] and Fig 5).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize Omoigui method of returning results from the operations performed on the database created by Bata. One would have been motivated to do so in order to provide results from queries to a user.

Referring to claim 23, Bata/Omoigui discloses the system of claim 22, wherein the remote service comprises an application available via a network (Bata: see column 4, lines 33-53 and Fig 3 – the service is available via the internet).

Referring to claim 24, Bata/Omoigui discloses the system of claim 22, wherein the defined interface comprises a Web Services Description Language (WSDL) interface (Omoigui: see [0467], lines 8-10 and [0532], lines 1-2).

Referring to claim 25, Bata/Omoigui discloses the system of claim 24, wherein said mapping module creates the database tables based on the WSDL interface (Omoigui: see [0467], lines 8-10 and [0532], lines 1-2).

Referring to claim 26, Bata/Omoigui discloses the system of claim 22, wherein said mapping module creates meta data identifying a particular method of the remote service to be invoked when an operation is received on a given database table (Omoigui: see [0736]).

Referring to claim 27, Bata/Omoigui discloses the system of claim 22, wherein said mapping module maps arguments of a method to columns of a database table (Bata: see column 9, lines 48-57 and Fig 9).

Referring to claim 28, Bata/Omoigui discloses the system of claim 22, wherein each database table created by the mapping module represents a method of the remote service (Bata: see column 9, lines 6-34).

Referring to claim 29, Bata/Omoigui discloses the system of claim 22, wherein said mapping module creates an object encapsulating the mapping of a method of the remote service to a database table (Bata: see column 6, lines 29-57 – the mapping is saved in a file folder).

Referring to claim 30, Bata/Omoigui discloses the system of claim 22, further comprising: a mapping repository for storing mappings between database tables and methods of the remote service (Bata: see column 6, lines 29-57 – hierarchy of file folder).

Referring to claim 31, Bata/Omoigui discloses the system of claim 30, wherein the conversion module consults the mapping repository for converting result values into database format (Bata: see column 7, lines 44-46).

Referring to claim 32, Bata/Omoigui discloses the system of claim 22, wherein the operation received on the database table comprises a selected one

of a SELECT operation, an INSERT operation, a JOIN operation, and a UNION operation (Omoigui: [0287], lines 12-15 – SELECT operation).

Referring to claim 33, Bata/Omoigui discloses the system of claim 22, wherein said invocation module binds the data from the operation to a Simple Object Access Protocol (SOAP) call for invoking the method of the remote service (Omoigui: see [0312] and [0362]).

Referring to claim 34, Bata/Omoigui discloses the system of claim 22, wherein said invocation module converts data from the database operation into Extensible Markup Language (XML) format (Bata: see column 3, lines 9-15 and Fig 9).

Referring to claim 35, Bata/Omoigui discloses the system of claim 22, wherein said invocation module creates a Simple Object Access Protocol (SOAP) request for invoking the method of the remote service (Omoigui: see [0312] and [0362]).

Referring to claim 36, Bata/Omoigui discloses the system of claim 35, wherein said communication module sends the SOAP request to the remote service (Omoigui: see [0312] and [0362]).

Referring to claim 37, Bata/Omoigui discloses the system of claim 22, wherein said conversion module converts result values received in Simple Object Access Protocol (SOAP) format into database data types (Omoigui: see [0312] and [0362]).

Referring to claim 38, Bata/Omoigui discloses the system of claim 22, wherein said conversion module converts result values received in Extensible

Markup Language (XML) format into database data types (Bata: see column 3, lines 9-15).

Referring to claim 39, Bata/Omoigui discloses the system of claim 22, wherein said conversion module provides converted result values in response to the operation on the database table (Bata: see column 9, lines 48-57 and Fig 9).

Referring to claim 40, Bata discloses a method of performing queries available from an application. In particular, Bata discloses in a database system, a method for performing database queries on data available from an application (see abstract), the method comprising:

- establishing communication between a database and an application having an interface (see Fig 1 – the internet provides communication);
- creating database tables to represent at least some functions of the application based on the interface, each database table corresponding to a function of the application (see column 9, lines 6-34);
- in response to a database query received on a database table corresponding to a function of the application, generating input arguments expected by the function based on the database query (see column 7, lines 35-43);
- invoking the function with the input arguments and receiving results from invoking the function (see column 10, lines 10-16 – the data is inserted);
- converting the results into a database result set (see column 8, lines 35-49 and Fig 7); and
- returning the database result set in response to the database query.

However, while Bata discloses a method for performing database operations on data obtained from a web service, Bata fails to explicitly teach the further limitation of returning result values. Omoigui teaches a similar method to that of Bata, including returning result values (see [0080] and Fig 5).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize Omoigui method of returning results from the operations performed on the database created by Bata. One would have been motivated to do so in order to provide results from queries to a user.

Referring to claim 41, Bata/Omoigui discloses the method of claim 40, wherein the application comprises a web service (Bata: see column 4, lines 42-46 and Fig 3, item 145H).

Referring to claim 42, Bata/Omoigui discloses the method of claim 40, wherein the application comprises a service available via a network (Bata: see column 4, lines 33-53 and Fig 3 – the service is available via the internet).

Referring to claim 43, Bata/Omoigui discloses the method of claim 40, wherein the interface comprises a Web Services Description Language (WSDL) interface (Omoigui: see [0467], lines 8-10 and [0532], lines 1-2).

Referring to claim 44, Bata/Omoigui discloses the method of claim 40, wherein said step of creating database tables includes creating meta data identifying a particular function to be invoked when an operation is received on a given database table (Omoigui: see [0736]).

Referring to claim 45, Bata/Omoigui discloses the method of claim 40, wherein said step of creating database tables includes mapping arguments of a

given function to columns of the corresponding database table (Bata: see column 9, lines 48-57 and Fig 9).

Referring to claim 46, Bata/Omoigui discloses the method of claim 40, wherein said step of invoking the function includes binding data from the database query to a Simple Object Access Protocol (SOAP) call (Omoigui: see [0312] and [0362]).

Referring to claim 47, Bata/Omoigui discloses the method of claim 40, wherein said step of invoking the function includes converting data from the database query into Extensible Markup Language (XML) format (Bata: see column 3, lines 9-15 and Fig 9).

Referring to claim 48, Bata/Omoigui discloses the method of claim 40, wherein said step of invoking the function includes creating a Simple Object Access Protocol (SOAP) request for invoking the function (Omoigui: see [0312] and [0362]).

Referring to claim 49, Bata/Omoigui discloses the method of claim 48, wherein said step of invoking the function includes transmitting the SOAP request to a remote server (Omoigui: see [0312] and [0362]).

Referring to claim 50, Bata/Omoigui discloses the method of claim 40, wherein said step of invoking the function includes receiving results in Extensible Markup Language (XML) format (Bata: see column 3, lines 9-15 and Fig 9).

Referring to claim 51, Bata/Omoigui discloses the method of claim 40, wherein said step of invoking the function includes receiving results in Simple Object Access Protocol (SOAP) format (Omoigui: see [0312] and [0362]).

Referring to claim 52, Bata/Omoigui discloses the method of claim 40, wherein said step of converting the results includes converting results received in Simple Object Access Protocol (SOAP) format (Omoigui: see [0312] and [0362]).

Referring to claim 53, Bata/Omoigui discloses the method of claim 40, wherein said step of converting the results includes converting results received in Extensible Markup Language (XML) format (Bata: see column 3, lines 9-15 and Fig 9).

Referring to claim 54, Bata/Omoigui discloses a computer-readable medium having processor-executable instructions for performing the method of claim 40 (Omoigui: see [0275]).

Referring to claim 55, Bata/Omoigui discloses a downloadable set of processor-executable instructions for performing the method of claim 40 (Omoigui: see [0275]).

Contact Information

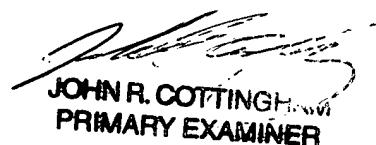
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kimberly Lovel whose telephone number is (571) 272-2750. The examiner can normally be reached on 8:00 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kimberly Lovel
Examiner
Art Unit 2167

kml
27 April 2006



JOHN R. COTTINGHAM
PRIMARY EXAMINER

